In this activity, you will create a NiFi pipeline for a Cassandra database. It will allow you to write data from a Cassandra database into a MySQL database. You will begin by populating a Cassandra database, and then you will build a NiFi pipeline that will allow you to perform some basic ETL operations between the Cassandra and MySQL databases. Finally, you will check that the data has been correctly written into your MySQL database.

Before starting this activity, review the submission instructions below to ensure that you collect the required screenshots as you progress through the activity.

Note that this activity has been tested using a Windows OS and the Catalina version of a Mac OS. If you use the Big Sur OS, you are recommended to use the myPhpAdmin *container* as demonstrated in this article: [Run MySQL & phpMyAdmin Locally Using Docker.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

[Links to an external site.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

**Reference**

[Yuste, Miguel. “Run MySQL & PhpMyAdmin Locally in 3 Steps Using Docker.” *Medium*. 2019. https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

[Links to an external site.](https://migueldoctor.medium.com/run-mysql-phpmyadmin-locally-in-3-steps-using-docker-74eb735fa1fc)

**To complete this activity, follow these steps:**

1. Create the NifiNetwork Docker network and the NiFi and Cassandra *containers* by following the steps in [Mini-Lesson 17.6](https://classroom.emeritus.org/courses/10605/pages/mini-lesson-17-dot-6-using-nifi-to-create-a-pipeline-in-cassandra-30-00).  
   Provide a screenshot of your Docker desktop to show that you successfully created the NiFi and Cassandra *containers* in the Docker network.
2. From the Docker desktop, start a bash window for the Cassandra *server*. Run the cqlsh command to run *queries* against the Cassandra database.  
   Provide a screenshot of the bash window in the Cassandra *server* to show that you successfully ran the cqlsh command.
3. Copy the code below into the Cassandra bash shell to create a data table for the Cassandra database and insert “Peter Parker” into the “person” table:

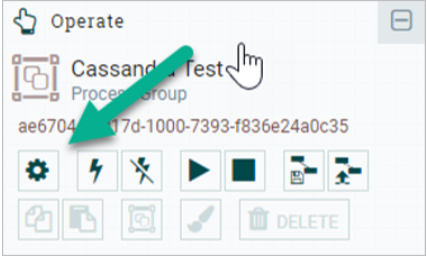
| CREATE KEYSPACE IF NOT EXISTS k1 WITH replication = {'class': 'SimpleStrategy', 'replication\_factor': '1'} AND durable\_writes = true; use k1;  CREATE TABLE person (  id text,  name text,  surname text,  PRIMARY KEY (id)); INSERT INTO person (id, name, surname) VALUES ('001', 'Peter', 'Parker'); Select \* from person; |
| --- |

Provide a screenshot of the Terminal window to show that “Peter Parker” was inserted correctly into the Cassandra data table.

4. Navigate to https://localhost:8080/nifi to start the NiFi UI. Provide a screenshot to show that you successfully started the NiFi UI.

5. Create aprocess group and name it Cassandra-test. Provide a screenshot to show that you successfully created the Cassandra-test process group.

6. Inside the Cassandra-test process group, select the gear icon to configure the Cassandra-test process group:



In the CONTROLLER SERVICES tab, select the plus sign, select CassandraSessionProvider as the type, and then select the gear icon. In the PROPERTIES tab, set the Cassandra Contact Points field equal to some-cassandra:9042. Set the Client Auth field equal to NONE and the Keyspace field equal to k1.  
Provide a screenshot to show that you correctly set the fields of the PROPERTIES tab.

7. Enable the CassandraSessionProvider *controller*. Provide a screenshot to show that the CassandraSessionProvider *controller* service was successfully activated.

8. Drop a *processor* onto the NiFi canvas and filter for “Cassandra”. Select the QueryCassandra type.  
Configure the PROPERTIES tab as follows: set the Cassandra Connection Provider field equal to CassandraSessionProvider, the Client Auth field equal to NONE, the Keyspace field equal to k1, and the Output Format field equal to JSON.  
Provide a screenshot to show that you correctly set the fields of the PROPERTIES tab.

9. Create a MySQL *connector* by running the following command in the Terminal window:  
​​docker run -p 3306:3306 --name mysql -e MYSQL\_ROOT\_PASSWORD=mypass --network NifiNetwork -d mysql:8.0  
Next, open up the connection using MySQL Workbench and run the following commands to generate a person table in the people database by running the following *queries*:

| DROP DATABASE IF EXISTS people;  CREATE DATABASE IF NOT EXISTS people;  USE people;  CREATE TABLE person (  id varchar(50),  name varchar(50),  surname varchar(50),  PRIMARY KEY (id)  ); select \* from person; |
| --- |

Provide a screenshot to show that the person table is empty within MySQL Workbench.

10. Connect the MySQL *container* to NiFi as demonstrated in [Video 17.7](https://classroom.emeritus.org/courses/10605/pages/creating-an-etl-pipeline-19-10) by using a DBCPConnectionPool *connector*. The *driver* must also be configured as demonstrated in [Video 17.6](https://classroom.emeritus.org/courses/10605/pages/creating-an-etl-pipeline-19-10). Provide a screenshot of the DBCPConnectionPool *connector* service activated within the NiFi UI.

11. Add a SplitJSON *processor* to the pipeline connected to the QueryCassandra *processor*. In the SETTINGS tab, select failure and original to Automatically Terminate Relationships. Configure the PROPERTIES tab by setting the JsonPath Expression equal to $.\*.  
Provide two screenshots. The first screenshot should show that you set the values in the SETTINGS tab correctly. The second screenshot should show that you updated the scheduling time in the PROPERTIES tab correctly.

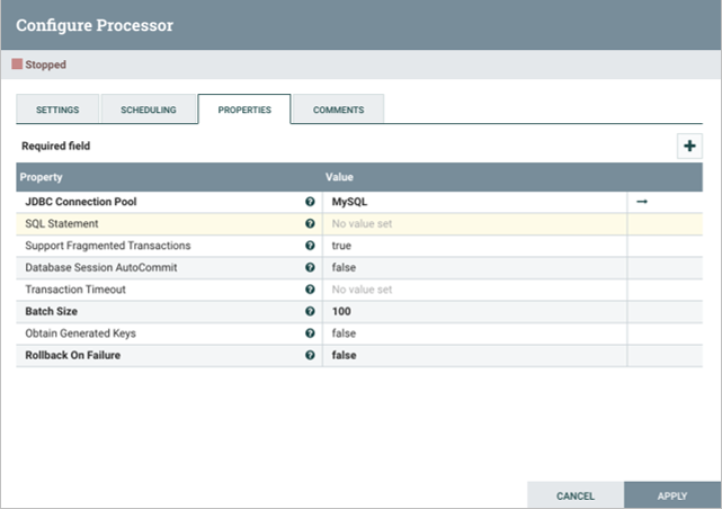
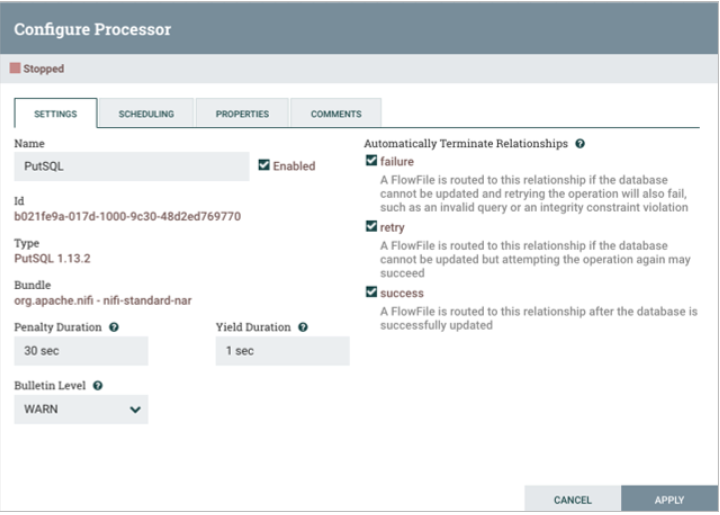
12. Add a ConvertJSONToSQL *processor* and connect it to the SplitJSON *processor* with splits as the relationship type. In the SETTINGS tab, select failure and original to Automatically Terminate Relationships. In the PROPERTIES tab set the following fields:

* 1. Set the JDBC Connection Pool field equal to MySQL.
  2. Set the Statement Type field equal to INSERT.
  3. Set the Table Name field equal to person.
  4. Set the Catalog Name field equal to people.
  5. Set the Translate Field Names field equal to false.
  6. Set the Output Format field equal to JSON.
  7. Set the SQL Parameter Attribute Prefix field equal to SQL.

Provide a screenshot to show that you configured the PROPERTIES tab for the ConvertJSONToSQL *processor* correctly.

13. Connect the QueryCassandra, SplitJSON, and ConvertJSONToSQL *processors*. The QueryCassandra and SplitJSON *processors* will be connected by a success relationship. The SplitJSON and ConvertJSONToSQL *processors* will be connected by a splits relationship. Provide a screenshot of the connected QueryCassandra, SplitJSON, and ConvertJSONToSQL *processors*.

14. Add a PutSQL *processor.* In the SETTINGS tab, select failure, retry, and success to Automatically Terminate Relationships. In the PROPERTIES tab, set the JDBC Connection Pool field equal to MySQL.



Connect the ConvertJSONToSQL and PutSQL *processors* by an sql relationship. Provide a screenshot to show that all four *processors* (QueryCassandra, SplitJSON, ConvertJSONToSQL, and PutSQL) are connected.

15. Run each *processor* and watch the data propagate through the pipeline. Finally, navigate back to the MySQL Workbench and run the following *query*:  
SELECT \* FROM person;  
Provide a screenshot to show the data was added to your MySQL database correctly.

**Submission Instructions:**

Your submission for this activity should be a Word document that includes the following screenshots, each labeled for the step that the screenshot represents:

1. Provide a screenshot of your Docker desktop to show that you successfully created the NiFi and Cassandra *containers* in the Docker network.
2. Provide a screenshot of the bash window in the Cassandra *server* to show that you successfully ran the cqlsh command.
3. Provide a screenshot of the Terminal window to show that “Peter Parker” was inserted correctly into the Cassandra data table.
4. Provide a screenshot to show that you successfully started the NiFi UI.
5. Provide a screenshot to show that you successfully created the Cassandra-test process group.
6. Provide a screenshot to show that you correctly set the fields of the PROPERTIES tab.
7. Provide a screenshot to show that the CassandraSessionProvider *controller* service was successfully activated.
8. Provide a screenshot to show that you correctly set the fields of the PROPERTIES tab.
9. Provide a screenshot to show that the person table is empty within MySQL Workbench.
10. Provide a screenshot of the DBCPConnectionPool *connector* service activated within the NiFi UI.
11. Provide two screenshots. The first screenshot should show that you set the values in the SETTINGS tab correctly. The second screenshot should show that you updated the scheduling time in the PROPERTIES tab correctly.
12. Provide a screenshot to show that you configured the PROPERTIES tab for the ConvertJSONToSQL *processor* correctly.
13. Provide a screenshot of the connected QueryCassandra, SplitJSON, and ConvertJSONToSQL *processors*.
14. Provide a screenshot to show that all four *processors* (QueryCassandra, SplitJSON, ConvertJSONToSQL, and PutSQL) are connected.
15. Provide a screenshot to show the data was added to your MySQL database correctly.